CLAIMS

1. A method of processing a computer graphics illustration, the computer graphics illustration having overlapping objects, comprising:

breaking up, without regard to the overlapping objects, at least a region of the computer graphics illustration into tiles having overlapping boundaries; and

planarizing the objects in each tile to create flattened vector output representing the tiled region of the computer graphics illustration.

2. The method of claim 1, wherein at least one of the overlapping objects is transparent and planarizing comprises:

converting the objects in each tile to an opaque flattened vector output representing the tiled region of the computer graphics illustration.

3. The method of claim 1, wherein breaking up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprises:

breaking up the computer graphics illustration into tiles having overlapping boundaries where a plurality of the tiles have a same shape.

4. The method of claim 1, wherein breaking up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprises:

breaking up the computer graphics illustration into tiles having overlapping boundaries where a plurality of the tiles have a same size.

5. The method of claim 1, wherein breaking up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprises:

breaking up the computer graphics illustration into square shaped tiles with overlapping boundaries.

6. The method of claim 1, wherein breaking up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprises:

breaking up the at least one region of the computer graphics illustration into a grid of adjacent, non-overlapping tiles; and

expanding the boundaries of one or more tiles in the grid in a direction such that the boundaries of the tiles overlap.

7. The method of claim 6, wherein expanding comprises:

expanding boundaries of one or more tiles in the grid by at least one pixel in the horizontal direction.

8. The method of claim 1, wherein breaking up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprises:

breaking up the at least one region of the computer graphics illustration into a grid of adjacent, non-overlapping tiles; and

expanding the boundaries of one or more tiles in the grid in a vertical direction such that the boundaries of the tiles overlap.

9. The method of claim 8, wherein expanding comprises:

expanding boundaries of one or more tiles in the grid by at least one pixel in the vertical direction.

10. The method of claim 1, further comprising:

selecting the size of the tiles based on an available amount of memory for planarizing the tiles.

11. The method of claim 10, further comprising:

determining if the available amount of memory for planarizing the tiles will be exceeded; and

dividing one or more of the overlapping tiles into smaller tiles with overlapping boundaries if the determining step indicates that the available amount of memory for planarizing the tiles will be exceeded.

12. The method of claim 11, wherein the determining step comprises:

predicting if the available amount of memory for planarizing the tiles will be exceeded.

13. The method of claim 11, wherein the determining step comprises:

monitoring if a memory overflow condition occurs during the planarizing of the objects in each tile.

14. The method of claim 11, further comprising:

repeating the dividing step until the size of each tile is such that the available amount of memory for planarizing each tile will not be exceeded.

15. The method of claim 10, further comprising:

selecting the size of the tiles based on a number of objects in a tile.

16. The method of claim 15, further comprising:

determining if the number of objects in the tile exceeds a threshold value; and

dividing one or more of the tiles with overlapping boundaries into smaller tiles with overlapping boundaries if the number of objects in the tile exceeds the threshold value.

17. The method of claim 16, further comprising:

repeating the dividing step until the size of each tile is such that the number of objects in each tile does not exceed the threshold value.

18. The method of claim 1, wherein breaking up at least a region comprises breaking up all of the computer graphics illustration.

19. The method of claim 1, further comprising:

aligning the border of each tile to physical boundaries of pixels in an output device space to make each tile include only pixels of the output device space that are wholly within the tile.

20. A method of processing a computer graphics illustration, the computer graphics illustration having overlapping objects, comprising:

providing, without regard to the overlapping objects, tiles having boundaries that overlap by an amount corresponding to at least one pixel in a device space, the tiles covering at least a region of the computer graphics illustration that includes overlapping objects;

supplying a clipping path corresponding to the boundary of each tile;

processing the objects surrounded by each clipping path to produce a vector output;

defining an alignment tool to be passed as a part of a stream to an output device, the alignment tool being operable to instruct an output device to align each clipping path to physical boundaries of pixels in the output device space to make each tile include only pixels of the output device space that are wholly within the tile; and

providing the tool and the vector output as a stream to the output device.

21. The method of claim 20, wherein the stream to the output device is a Postscript™ stream.

22. The method of claim 20, wherein at least one of the overlapping objects is transparent and processing comprises:

converting the objects in each tile to an opaque flattened vector output representing a tiled region of the computer graphics illustration.

23. The method of claim 20, wherein providing tiles having overlapping boundaries comprises:

providing a plurality of tiles having a same shape.

24. The method of claim 20, wherein providing tiles having overlapping boundaries comprises:

providing a plurality of the tiles having a same size.

25. The method of claim 20, wherein providing tiles having overlapping boundaries comprises:

providing a grid of adjacent, non-overlapping tiles; and

expanding the boundaries of one or more tiles in the grid in a direction such that the boundaries of the tiles overlap.

26. The method of claim 20, wherein providing tiles having overlapping boundaries comprises:

providing a grid of adjacent, non-overlapping tiles; and

expanding the boundaries of one or more tiles in the grid in a horizontal direction such that the boundaries of the tiles overlap.

27. The method of claim 20, wherein providing tiles having overlapping boundaries comprises:

providing a grid of adjacent, non-overlapping tiles; and

expanding the boundaries of one or more tiles in the grid in a vertical direction such that the boundaries of the tiles overlap.

28. The method of claim 20, further comprising:

selecting the size of the tiles based on an available amount of memory for processing the tiles.

29. The method of claim 28, further comprising:

determining if the available amount of memory for processing the tiles will be exceeded; and

dividing one or more of the tiles into smaller tiles with overlapping boundaries if the determining step indicates that the available amount of memory for processing the tiles will be exceeded.

30. The method of claim 29, further comprising:

repeating the dividing step until a size of each tile is such that the available amount of memory for processing each tile will not be exceeded.

31. The method of claim 28, further comprising:

selecting the size of the tiles based on a number of objects in a tile.

32. The method of claim 31, further comprising:

determining if the number of objects in the tile exceeds a threshold value; and

dividing one or more of the tiles with overlapping boundaries into smaller tiles with overlapping boundaries if the number of objects in the tile exceeds the threshold value.

33. The method of claim 32, further comprising:

repeating the dividing step until the size of each tile is such that the number of objects in each tile does not exceed the threshold value.

34. A method of processing a computer graphics illustration, the computer graphics illustration having overlapping objects wherein at least one object is transparent, comprising:

selecting a region of the computer graphics illustration to be processed, the region comprising at least a portion of the transparent object;

creating, without regard to the overlapping objects, a grid of overlapping tiles covering the selected region; and

supplying a clipping path corresponding to the boundary of each tile;

defining an alignment tool to be passed as a part of a stream to an output device, the alignment tool being operable to instruct the output device to align each clipping path to physical boundaries of pixels in an output device space;

for each tile in the tile grid:

processing the objects surrounded by each clipping path to produce a vector output;

passing the vector output and the alignment tool in a stream to the output device;

using the alignment tool to align each tile to physical boundaries of pixels in the output device space to make each tile include only pixels of the output device space that are wholly within the tile; and

rendering the processed objects as vector data on the output device.

35. The method of claim 34, wherein creating a grid of overlapping tiles comprises:

creating a grid of overlapping tiles in which adjacent tiles overlap by at least one pixel.

36. The method of claim 34, wherein at least one of the overlapping objects is transparent and processing comprises:

converting the objects in each tile to an opaque flattened vector output representing the tiled region of the computer graphics illustration.

37. A method of processing a computer graphics illustration, the computer graphics illustration having overlapping objects, comprising:

obtaining information about physical pixel boundaries in an output device space;

providing, based on the pixel boundary information and without regard to the overlapping objects, adjacent tiles covering at least a region of the computer graphics illustration that includes overlapping objects;

supplying a clipping path corresponding to the boundary of each tile;

processing the objects surrounded by each clipping path to produce vector output; and

passing the vector output as a stream to an output device.

38. A computer program product for processing a computer graphics illustration, comprising instructions operable to cause a programmable processor to:

break up, without regard to the overlapping objects, at least a region of the computer graphics illustration into tiles having overlapping boundaries; and

planarize the objects in each tile to create flattened vector output representing the tiled region of the computer graphics illustration.

39. The computer program product of claim 38, wherein at least one of the overlapping objects is transparent and the instructions to planarize comprise instructions to:

convert the objects in each tile to an opaque flattened vector output representing the tiled region of the computer graphics illustration.

40. The computer program product of claim 38, wherein the instructions to break up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprise instructions to:

break up the computer graphics illustration into tiles having overlapping boundaries where a plurality of the tiles have a same shape.

41. The computer program product of claim 38, wherein the instructions to break up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprise instructions to:

break up the computer graphics illustration into tiles having overlapping boundaries where a plurality of the tiles a the same size.

42. The computer program product of claim 38, wherein instructions to break up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprise instructions to:

break up the computer graphics illustration into square shaped tiles with overlapping boundaries.

43. The computer program product of claim 38, wherein the instructions to break up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprise instructions to:

break up the at least one region of the computer graphics illustration into a grid of adjacent, non-overlapping tiles; and

expand the boundaries of one or more tiles in the grid in a direction such that the boundaries of the tiles overlap.

44. The computer program product of claim 43, wherein the instructions to expand comprise instructions to:

expand boundaries of one or more tiles in the grid by at least one pixel in the horizontal direction.

45. The computer program product of claim 38, wherein the instructions to break up at least a region of the computer graphics illustration into tiles having overlapping boundaries comprise instructions to:

break up the at least one region of the computer graphics illustration into a grid of adjacent, non-overlapping tiles; and

expand the boundaries of one or more tiles in the grid in a vertical direction such that the boundaries of the tiles overlap.

46. The computer program product of claim 45, wherein the instructions to expand comprise instructions to:

expand boundaries of one or more tiles in the grid by at least one pixel in the vertical direction.

47. The computer program product of claim 38, further comprising instructions to:

select the size of the tiles based on an available amount of memory for planarizing the tiles.

48. The computer program product of claim 47, further comprising instructions to:

determine if the available amount of memory for planarizing the tiles will be exceeded; and

divide one or more of the overlapping tiles into smaller tiles with overlapping boundaries if the determination indicates that the available amount of memory for planarizing the tiles will be exceeded.

49. The computer program product of claim 48, wherein the instructions to determine comprise instructions to:

predict if the available amount of memory for planarizing the tiles will be exceeded.

50. The computer program product of claim 48, wherein the instructions to determine comprise instructions to:

monitor if a memory overflow condition occurs during the planarizing of the objects in each tile.

51. The computer program product of claim 48, further comprising instructions to:

repeat the dividing step until the size of each tile is such that the available amount of memory for planarizing each tile will not be exceeded.

52. The computer program product of claim 47, further comprising instructions to:

select the size of the tiles based on the number of objects in a tile.

53. The computer program product of claim 52, further comprising instructions to:

determine if the number of objects in the tile exceeds a threshold value; and

divide one or more of the tiles with overlapping boundaries into smaller tiles with overlapping boundaries if the number of objects in the tile exceeds the threshold value.

54. The computer program product of claim 53, further comprising instructions to:

repeat the dividing step until the size of each tile is such that the number of objects in each tile does not exceed the threshold value.

55. The computer program product of claim 38, wherein the instructions to break up at least a region comprise instructions to break up all of the computer graphics illustration.

56. The computer program product of claim 38, further comprising instructions to:

align the border of each tile to physical boundaries of pixels in an output device space to make each tile include only pixels of the output device space that are wholly within the tile.

57. A computer program product for processing a computer graphics illustration, comprising instructions operable to cause a programmable processor to:

provide, without regard to the overlapping objects, tiles having boundaries that overlap by an amount corresponding to at least one pixel in a device space, the tiles covering at least a region of the computer graphics illustration that includes overlapping objects;

supply a clipping path corresponding to the boundary of each tile;

process the objects surrounded by each clipping path to produce a vector output;

define an alignment tool to be passed as a part of a stream to an output device, the alignment tool being operable to instruct the output device to align each clipping path to physical boundaries of pixels in the output device space to make each tile include only pixels of the output device space that are wholly within the tile; and

provide the tool and the vector output as a stream to the output device

58. The computer program product of claim 57, wherein the stream to the output device is a Postscript™ stream.

59. The computer program product of claim 57, wherein at least one of the overlapping objects is transparent and the instructions to process comprises instructions to:

convert the objects in each tile to an opaque flattened vector output representing a tiled region of the computer graphics illustration.

60. The computer program product of claim 57, wherein the instructions to provide tiles having overlapping boundaries comprises instructions to:

provide a plurality of the tiles having a same shape.

61. The computer program product of claim 57, wherein the instructions to provide tiles having overlapping boundaries comprises instructions to:

provide a plurality of the tiles having a same size.

62. The computer program product of claim 57, wherein the instructions to provide tiles having overlapping boundaries comprises instructions to:

provide a grid of adjacent, non-overlapping tiles; and

expand the boundaries of one or more tiles in the grid in a direction such that the boundaries of the tiles overlap.

63. The computer program product of claim 57, wherein the instructions to provide tiles having overlapping boundaries comprises instructions to:

provide a grid of adjacent, non-overlapping tiles; and

expand the boundaries of one or more tiles in the grid in a horizontal direction such that the boundaries of the tiles overlap.

64. The computer program product of claim 57, wherein the instructions to provide tiles having overlapping boundaries comprises instructions to:

provide a grid of adjacent, non-overlapping tiles; and

expand the boundaries of one or more tiles in the grid in a vertical direction such that the boundaries of the tiles overlap.

65. The computer program product of claim 57, further comprising instructions to:

select the size of the tiles based on an available amount of memory for processing the tiles.

66. The computer program product of claim 65, further comprising instructions to:

determine if the available amount of memory for processing the tiles will be exceeded; and

divide one or more of the tiles into smaller tiles with overlapping boundaries if the determination indicates that the available amount of memory for processing the tiles will be exceeded.

67. The computer program product of claim 66, further comprising instructions to:

repeat the dividing step until a size of each tile is such that the available amount of memory for processing each tile will not be exceeded.

68. The computer program product of claim 65, further comprising instructions to:

select the size of the tiles based on a number of objects in a tile.

69. The computer program product of claim 68, further comprising instructions to:

determine if the number of objects in the tile exceeds a threshold value; and

divide one or more of the tiles with overlapping boundaries into smaller tiles with overlapping boundaries if the number of objects in the tile exceeds the threshold value.

70. The computer program product of claim 69, further comprising instructions to:

repeat the dividing step until the size of each tile is such that the number of objects in each tile does not exceed the threshold value.

71. A computer program product for processing a computer graphics illustration, comprising instructions operable to cause a programmable processor to:

select a region of the computer graphics illustration to be processed, the region comprising at least a portion of the transparent object;

create, without regard to the overlapping objects, a grid of overlapping tiles covering the selected region; and

supply a clipping path corresponding to the boundary of each tile;

define an alignment tool to be passed as a part of a stream to an output device, the alignment tool being operable to instruct the output device to align each clipping path to physical boundaries of pixels in an output device space;

for each tile in the tile grid:

process the objects surrounded by each clipping path to produce a vector output;

pass the vector output and the alignment tool in a stream to the output device;

use the alignment tool to align each tile to physical boundaries of pixels in the output device space to make each tile include only pixels of the output device space that are wholly within the tile; and

render the processed objects as vector data on the output device.

72. The computer program product of claim 71, wherein the instructions to create a grid of overlapping tiles comprises instructions to:

create a grid of overlapping tiles in which adjacent tiles overlap by at least one pixel.

73. The computer program product of claim 71, wherein at least one of the overlapping objects is transparent and the instructions to process comprises instructions to:

convert the objects in each tile to an opaque flattened vector output representing the tiled region of the computer graphics illustration.

74. A computer program product for processing a computer graphics illustration, comprising instructions operable to cause a programmable processor to:

obtain information about physical pixel boundaries in an output device space;

provide, based on the pixel boundary information and without regard to the overlapping objects, adjacent tiles covering at least a region of the computer graphics illustration that includes overlapping objects;

supply a clipping path corresponding to the boundary of each tile;

process the objects surrounded by each clipping path to produce a vector output; and

pass the vector output as a stream to an output device.